Noise-contrastive estimation of normalising constants and GANs

Fonctions génériques

```
library(ggplot2)
library(reshape)
library(matrixStats)
```

Algorithme d'Hasting

Utilité : simuler selon $p_m(., \psi)$ pour un paramètre ψ choisi.

Argument	Type	Exemple	Indication
x	vecteur	rcauchy(100, 0, 1)	notre échantillon de densité inconnue
n	entier	100	taille de la simulation
psi	vecteur	c(0,1)	paramètres de la fonction h
h	fonction		fonction qui retourne $\overline{p_m}(., \psi)$

```
hasting = function(x, n, psi, h){
  y = c()
  y = append(y, sample(sample(x, 1)))
  for (i in 2:n){
    y_ = rnorm(1, y[i-1], 1)
    u = runif(1)
    if ( u <=
         (h(y_,psi) * dnorm(y_, y[i-1], 1))
         /(h(y[i-1],psi) * dnorm(y[i-1], y_, 1))
    ){
      y = append(y, y_)}
    else {
      y = append(y, y[i-1])
    }
  }
  return (y)
}
```

Note : on peut très certainement écrire sous forme matricielle cette fonction pour une meilleure performance.

MC MLE

Utilité : retourne une estimation des paramètres selon la méthode décrite dans le papier de Geyer.

```
mc_mle = function(x, psi, h){

  m = length(x)

  y = hasting(x, m, psi, h)
```

```
  L = function(theta){
    return(sum(log(h(x,theta)/h(x,psi))) - m*log(mean(h(y,theta)/h(y,psi))))
  }

  theta = optim(
    par = rep(1,length(psi)),
    gr = "CG",
    control = list(fnscale=-1),
    fn = L
  )$par

  return(theta)
}
```

NCE

Utilité : Retourne l'estimation de la constante et des paramètres.

Argument	Type	Exemple	Indication
x	vecteur	rcauchy(100, 0, 1)	notre échantillon de densité inconnue
law_y	fonction	rnorm	fonction qui retourne un échantillon suivant la loi p_n
n	entier	100	taille de l'échantillon de bruit suivant la loi p_n
params_y	vecteur	c(0,1)	arguments de la fonction law_y
log_pm	fonction		fonction qui retourne le logarithme de la densité p_m
log_pn	fonction		fonction qui retourne le logarithme de la densité p_n
size_theta	entier	3	taille de θ, vaut habituellement 2 ou 3
method	string	"CG"	méthode d'optimisation, habituellement "CG" ou "BFGS"

```
nce = function(x, law_y, params_y, log_pm, log_pn, size_theta, n, methode = "CG"){

  y = do.call(law_y, c(list(n),params_y))

  m = length(x)

  h = function(u, theta){
    return( 1 / (1 + n/m * exp(log_pn(u) - log_pm(u, theta))))
  }

  J = function(theta){
    return( sum(log(h(x, theta))) + sum(log(1 - h(y, theta))) )
  }

  theta = optim(
    par = rep(1, size_theta),
    gr = methode,
    control = list(fnscale=-1),
    fn = J
  )$par

  return(c(theta[-size_theta], exp(-theta[size_theta])))
}
```

Utilité : utilise le bootstrap sur x pour estimer les paramètres

```r
NCE_bootstrap = function(x, law_y, params_y, log_pm, log_pn, size_theta, n,  nb_of_boot, methode = "CG"

  theta_bootstrap = c()
  for (i in 1:nb_of_boot) {
    x_bootstrap = sample(x, size = m, replace=TRUE)
    theta_bootstrap = append(theta_bootstrap, nce(x_bootstrap, law_y, params_y, log_pm, log_pn, size_th
  }

  # Formatage des données
  theta_bootstrap = matrix(theta_bootstrap, nrow = size_theta)

  # Plot
  #print(t(rowCumsums(theta_bootstrap)) / 1:nb_of_boot)

  return(rowMeans(theta_bootstrap))
}
```

Graphiques

Utilité : afficher l'histogramme pour un échantillon de données x.

```r
print_hist = function(x) {
  df = data.frame(x = x)
  hist_x = ggplot(df, aes(x=x)) + geom_histogram(aes(y = stat(count) / sum(count)), bins = 20, color="w
  print(hist_x)
}
```

Utilité : pour NCE, afficher l'évolution des paramètres au fur et à mesure de l'augmentation de n (la dimension de l'échantillon de bruit)

```r
NCE_evol_params = function(x, law_y, params_y, log_pm, log_pn, size_theta, ratio, steps, labels, method

  # Creation de l'abscisse
  m = length(x)
  N = seq(0, m*ratio, length.out = steps + 1)

  # Creation de l'ordonnée
  theta = c()
  for (n in N) {
    theta = append(theta, nce(x, law_y, params_y, log_pm, log_pn, size_theta, n, methode))
  }

  # Formatage des données
  theta = t(rbind(matrix(theta, nrow = size_theta),N))
  df = as.data.frame(theta)
  df_melted = melt(df, id.vars = "N")

  # Plot
  plot_df = ggplot(df_melted, aes(x = N, y = value)) +
  geom_line(aes(color = variable, group = variable)) +
  geom_point(aes(color = variable, group = variable)) +
  labs(title = "Evolution des paramètres", x = "n", y = "Paramètres", color = "Légende") +
  scale_color_manual(labels = labels, values = c("blue", "red", "orange"))

  print(plot_df)
```

```
  return(theta)
}
```

Exemple basique : la loi normale

Soit x l'échantillon de taille m obtenu selon la loi de densité inconnue p_d.

On considère ici que p_d appartient à la famille de fonctions paramétrées par $\theta = (c, \mu, \sigma)$ suivante :

$$p_m(u;\theta) = \frac{1}{Z(\mu,\sigma)} \times exp\big[-\frac{1}{2}\big(\frac{u-\mu}{\sigma}\big)^2\big] \quad \text{d'où} \quad ln(p_m(u;\theta)) = c - \frac{1}{2}\big(\frac{u}{\sigma} - \frac{\mu}{\sigma}\big)^2$$

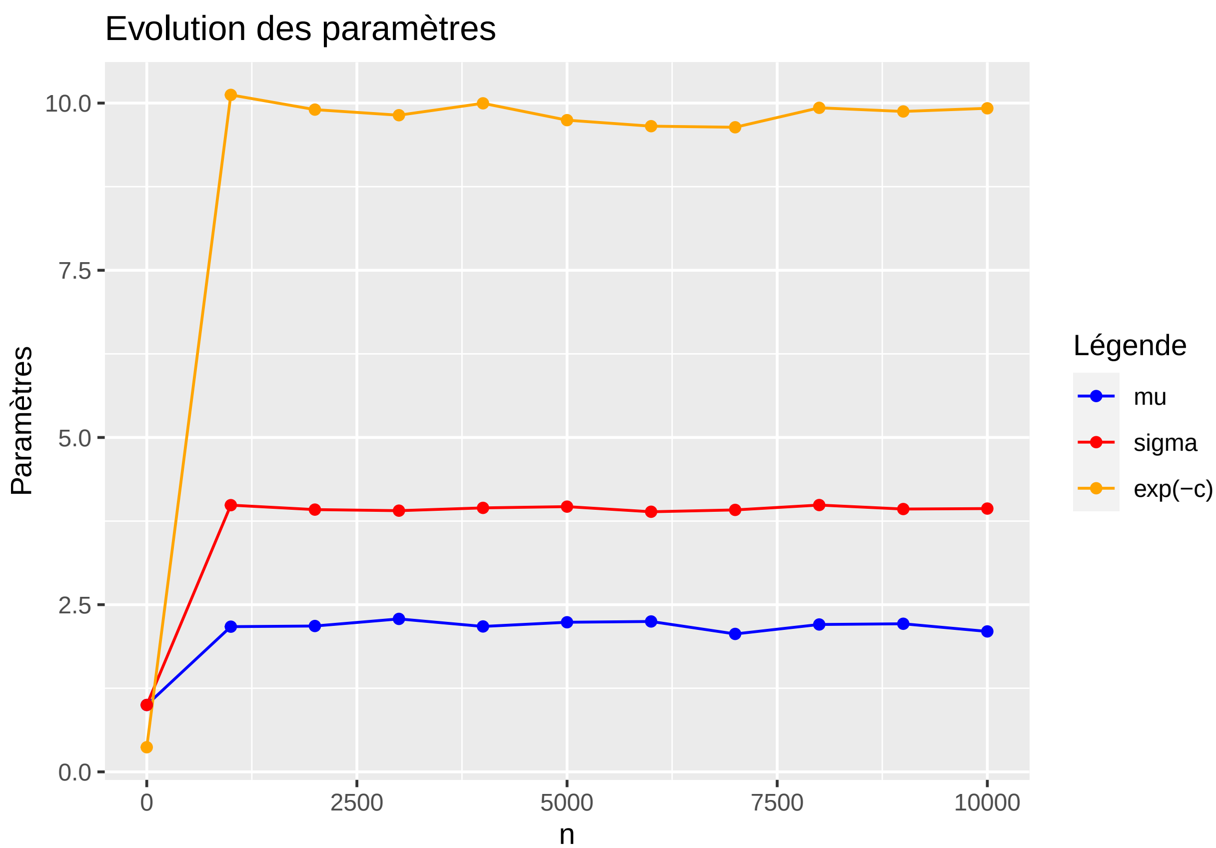
```
pm_barre = function(u, theta){
  return(exp(-0.5 * ((u - theta[1]) / theta[2]) ** 2))
}

log_pm = function(u,theta){
  return(theta[3] - 1/2 * (u/theta[2] - theta[1]/theta[2]) ** 2)
  # theta[1] = mu / theta[2] = sigma / theta[3] = c
}

log_pn_cauchy = function(u){
  return(log(dcauchy(u, mean(x), sd(x))))
}

m = 1000
n = 10000
x = rnorm(m, 2, 4)
size_theta = 3

# METHODE MC MLE
mc_mle(x, c(mean(x),sd(x)), pm_barre)
```

```
## [1]  192412.1 2067626.0
```

```
# METHODE GEYER
nce(x, rcauchy, c(mean(x),sd(x)), log_pm, log_pn_cauchy, size_theta, n)
```

```
## [1] 2.177080 3.964129 9.893548
```

```
NCE_evol_params(x, rcauchy, c(mean(x),sd(x)), log_pm, log_pn_cauchy, size_theta, 10, 10, c("mu", "sigma
```




```
##                                      N
##  [1,] 1.000000 1.000000  0.3678794     0
##  [2,] 2.171049 3.987867 10.1224673  1000
##  [3,] 2.180829 3.921300  9.9022075  2000
##  [4,] 2.286849 3.905458  9.8186336  3000
##  [5,] 2.174087 3.947022  9.9965753  4000
##  [6,] 2.236736 3.965814  9.7446465  5000
##  [7,] 2.248496 3.889123  9.6546146  6000
##  [8,] 2.062256 3.916697  9.6383770  7000
##  [9,] 2.203476 3.989269  9.9291988  8000
## [10,] 2.214197 3.929776  9.8748777  9000
## [11,] 2.099845 3.936656  9.9219186 10000
```

```
(NCE_bootstrap(x, rcauchy, c(mean(x),sd(x)), log_pm, log_pn_cauchy, size_theta, n, 10))
```

```
## [1] 2.185907 3.944246 9.882722
```